

# C LIBRARY FUNCTION - REALLOC

[http://www.tutorialspoint.com/c\\_standard\\_library/c\\_function\\_realloc.htm](http://www.tutorialspoint.com/c_standard_library/c_function_realloc.htm)

Copyright © tutorialspoint.com

## Description

The C library function **void \*realloc***void \* ptr, size\_t size* attempts to resize the memory block pointed to by **ptr** that was previously allocated with a call to **malloc** or **calloc**.

## Declaration

Following is the declaration for realloc function.

```
void *realloc(void *ptr, size_t size)
```

## Parameters

- **ptr** -- This is the pointer to a memory block previously allocated with malloc, calloc or realloc to be reallocated. If this is NULL, a new block is allocated and a pointer to it is returned by the function.
- **size** -- This is the new size for the memory block, in bytes. If it is 0 and ptr points to an existing block of memory, the memory block pointed by ptr is deallocated and a NULL pointer is returned.

## Return Value

This function returns a pointer to the newly allocated memory, or NULL if the request fails.

## Example

The following example shows the usage of realloc function.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char *str;

    /* Initial memory allocation */
    str = (char *) malloc(15);
    strcpy(str, "tutorialspoint");
    printf("String = %s, Address = %u\n", str, str);

    /* Reallocating memory */
    str = (char *) realloc(str, 25);
    strcat(str, ".com");
    printf("String = %s, Address = %u\n", str, str);

    free(str);

    return(0);
}
```

Let us compile and run the above program that will produce the following result:

```
String = tutorialspoint, Address = 355090448
String = tutorialspoint.com, Address = 355090448
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```